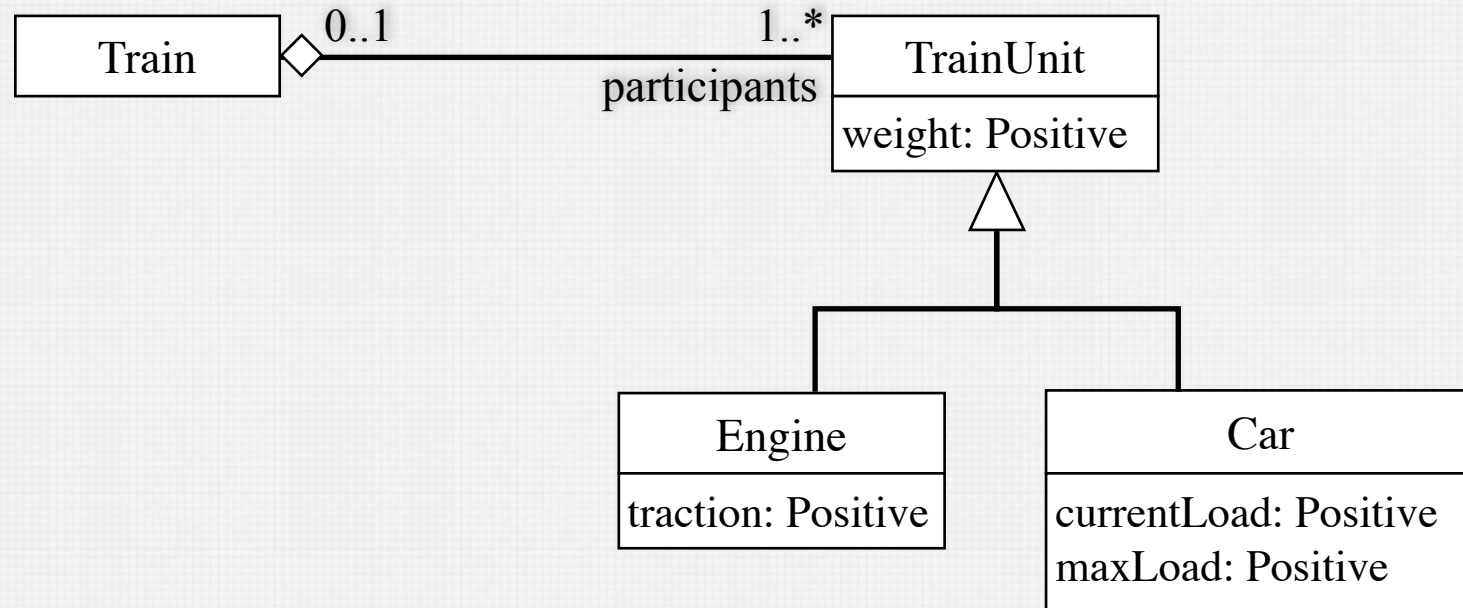


TRAIN DEPOT DOMAIN MODEL ANSWER



TRAIN DEPOT OCL ANSWERS (1)

- Write the following constraints and functions in OCL:
(if your model already models that constraint, then just write: “Is covered by model”)

1. The current load of a car cannot exceed its capacity.

context Car

inv: **self**.currentLoad \leq **self**.maxLoad

2. The length of a train should not exceed 25 train units,
i.e. cars or train engines.

context Train

inv: **self**.participant \rightarrow **size**() \leq 25

(or change the multiplicity in the model)

TRAIN DEPOT OCL ANSWERS (2)

3. Every train must have at least one train engine.

context Train

inv: self.participant → **exists**

(u : TrainUnit | u.**oclIsKindOf**(TrainEngine))

or

inv: self.participant → **select**

(**oclIsKindOf**(TrainEngine)) → **size()** ≥ 1

TRAIN DEPOT OCL ANSWERS (3)

4. Write an OCL function that computes the total weight of a train without considering the current load of the cars.

context Train

def: totalWeight() : Positive

= **self**.participant → **collect**(weight) → **sum**()

(or shortcut: **self**.participant.weight → **sum**())

TRAIN DEPOT ANSWERS (4)

5. Write an OCL function that computes the total traction strength of a train.

context TrainDepot::totalTraction(t : Train) :

Positive

post:

result = t.participant → **select**

(u : TrainUnit | u.**oclIsTypeOf**(TrainEngine))

→ **collect**(traction) → **sum**()

TRAIN DEPOT ANSWERS (5)

6. The total weight of a train plus the load in the cars cannot exceed the total traction strength of the engines of the train. (You are allowed to use the functions declared above.)

context t : Train

inv: totalTraction(t) \geq t.totalWeight() +
t.participant \rightarrow **select**(u : TrainUnit |
u.collsTypeOf
(Car)).currentLoad \rightarrow **sum**()

TRAIN DEPOT ANSWERS (6)

7. Write an OCL function that computes the available load of a train, respecting all invariants mentioned above (i.e. enough room, engines are strong enough).

context TrainDepot

def: availableLoad(t : Train) : Positive

= (totalTraction(t) - t.totalWeight(t) - t.participant → **select**
(**ocllsTypeOf**(Car)).currentLoad → **sum**()) **.min**
(t.participant → **select**(**ocllsTypeOf**(Car))
.maxLoad → **sum**() - t.participant → **select**
(**ocllsTypeOf**(Car)).currentLoad) → **sum**())

LIBRARY OCL ANSWERS (1)

1. A book cannot be borrowed by more than one member.

Covered by model

2. The number of books on loan for a given member does not exceed the maximum number of books on loan allowed for his category.

context Member:

inv : **self.borrowedBook** → **size()** ≤
self.memberCategory.maxNbBooks

LIBRARY OCL ANSWERS (2)

3. A given member cannot be twice on the waiting list for the same book.

Covered by model

4. A member is not allowed to place holds on more than 5 books in each category.

context LibrarySystem:

inv : **self**.bookCategory → **forall**(c |

self.member → **forall**(m | m.bookOnHold → **select**

(b | b.book.bookCategory = c) → **size**() ≤ 5))

LIBRARY OCL ANSWERS (3)

5. Every book category must have a maximum length of loan period defined for every member category.

context LibrarySystem:

inv : **self**.loanPeriod → **size**() =
self.bookCategory → **size**() * **self**.memberCategory → **size**()

6. A book that is on reserve can not be on loan.

context BookCopy:

inv : **self**.onReserve **implies**
(**self**.currentHolder → **isEmpty**())

LIBRARY OCL ANSWERS (4)

7. Write a function that calculates, for a given member, how long s/he is allowed to borrow a given book.

context LibrarySystem

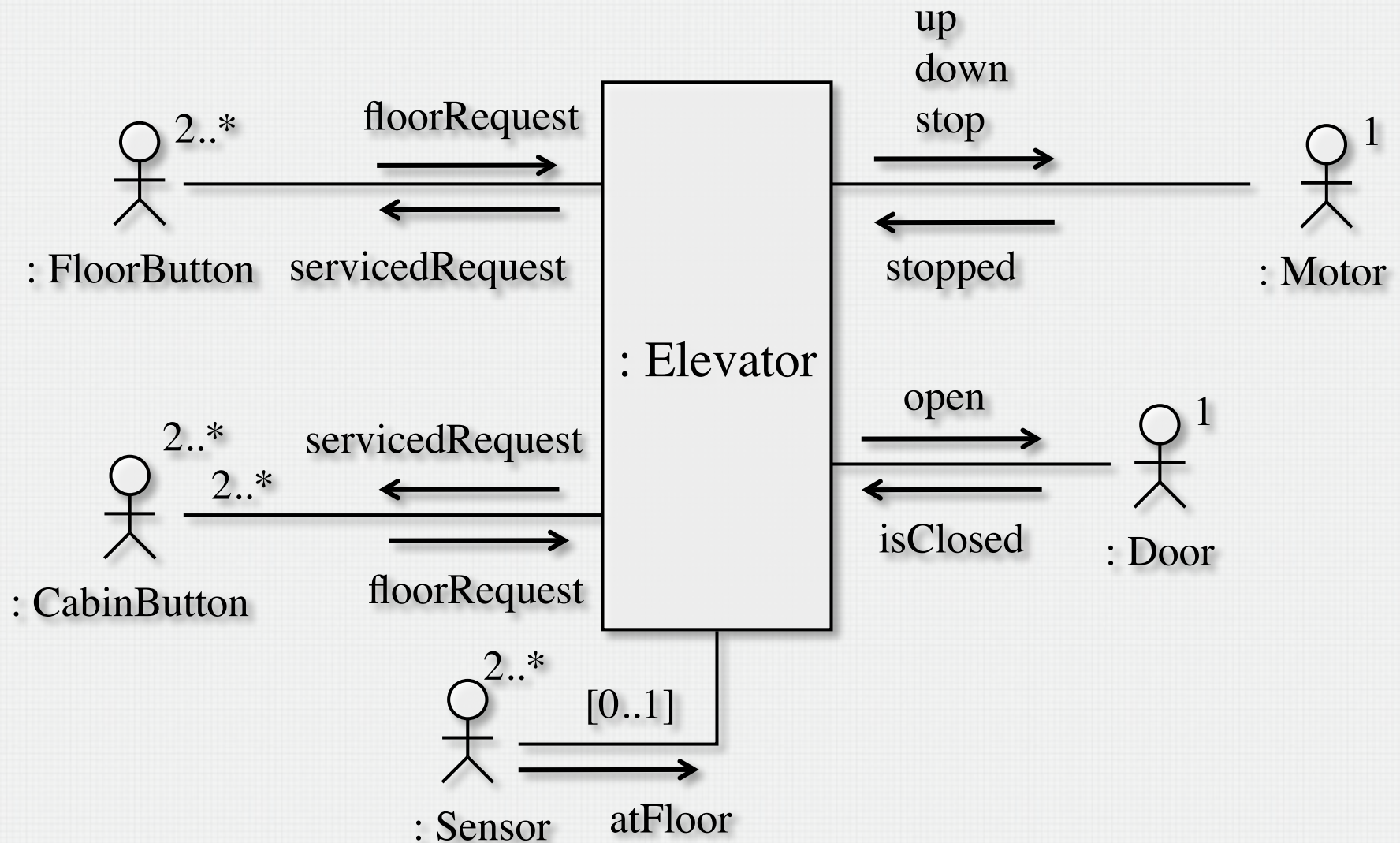
```
def lengthOfLoan(m: Member, b: Book) : Integer =  
self.loanPeriod → any(lp | lp.bookCategory = b.bookCategory and  
lp.memberCategory = m.memberCategory).duration  
(instead of any, select can be used)
```

8. Write a function that calculates, for a given member, when (date) he has to go to the library the next time (because one of his books on loan has to be returned).

context LibrarySystem:

```
def nextVisit(m: Member) : Integer = m.loan.endDate → any  
(md | m.loan.endDate → forAll(d | md ≤ d))
```

ELEVATOR ENVIRONMENT MODEL ANSWER



ELEVATOR CONCEPT MODEL ANSWER

